

Package: pprules (via r-universe)

October 15, 2026

Encoding UTF-8

Type Package

Title Piecepack Game Rules

Version 0.1.2-3

Description Functions to generate piecepack game rulesets and books.

License CC BY-SA 4.0

URL <https://github.com/piecepackr/pprules>,
<https://trevorldavis.com/R/pprules/>

BugReports <https://github.com/piecepackr/pprules/issues>

LazyLoad yes

Depends R (>= 3.4.0)

Imports dplyr, grid, grDevices, knitr, piecepackr (>= 1.10.1), ppdf,
R6, rlang, snakecase, stats, stringr, tibble, tools, utils,
withr, xmpdf (>= 0.1.0-32), yaml

Suggests systemfonts, testthat,

Remotes piecepackr/ppdf

SystemRequirements exiftool, pandoc, xelatex

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/pak/sysreqs libimage-exiftool-perl libicu-dev libjpeg-dev libpng-dev libxml2-dev

Repository <https://piecepackr.r-universe.dev>

Date/Publication 2026-02-05 18:38:53 UTC

RemoteUrl <https://github.com/piecepackr/pprules>

RemoteRef HEAD

RemoteSha ccf9eb301c5011daf57e0c9db439302cb987ebeb

Contents

game_kit	2
names_rulesets	2
save_promo_image	3
save_rulebook	4
save_ruleset	4
Index	7

game_kit	<i>Game Kit R6 object</i>
----------	---------------------------

Description

game_kit creates a game kit R6 object.

Usage

```
game_kit(cfgs = NULL)
```

Arguments

cfgs A named list of `piecepackr::pp_cfg()` configuration list objects. If NULL (default) we will use the "piecepack" configuration from `piecepackr::game_systems()`.

Examples

```
cfg <- piecepackr::game_systems()$dual_piecepacks_expansion
gk <- game_kit(list(cfg = cfg))
if (pprules::has_xelatex()) {
  output <- tempfile(fileext = ".pdf")
  save_pamphlet("tablut", gk = gk, output = output)
  # xopen::xopen(output)
  # browseURL(output)
}
```

names_rulesets	<i>Get names of piecepack games we can generate rulesets for.</i>
----------------	---

Description

names_rulesets() returns the names of piecepack games we can generate rulesets for.

Usage

```
names_rulesets(book = NULL)
```

Arguments

book Book name or NULL (for all supported rules). Currently only supports "the historical piecepacker".

See Also

[save_ruleset\(\)](#) and [save_pamphlet\(\)](#).

Examples

```
names_rulesets()
```

save_promo_image	<i>Save promo image of game</i>
------------------	---------------------------------

Description

save_promo_image() saves a promo image of a game. This is usually just a starting diagram for the game but sometimes we may create a custom image.

Usage

```
save_promo_image(game, gk = game_kit(), file = NULL, ...)
```

Arguments

game Game name. Will be normalized by [ppdf::normalize_name\(\)](#).
gk A game_kit R6 object.
file Filename for the image. If NULL we'll generate a pdf image with an appropriate name given game.
... Passed to [piecepackr::render_piece\(\)](#).

Details

If we haven't created a custom image for this game then basically we'll convert the game name to snake case, prepend a "piecepack_" in front and then `dynGet()` a function with that name and use the resulting data frame with [piecepackr::render_piece\(\)](#). See [ppdf::piecepack_games\(\)](#) for list of games directly supported by this package.

Value

A list with the width, height, and filename of the promo image. As a side effect we save an image to disk.

Examples

```
file <- tempfile(fileext = ".pdf")
whf <- save_promo_image("Fuji-san", gk = game_kit(), file = file)
print(whf)
unlink(whf$file)
```

save_rulebook	<i>Save rulebook</i>
---------------	----------------------

Description

save_rulebook saves a rulebook.

Usage

```
save_rulebook(
  book = "The Historical Piecepacker",
  gk = game_kit(),
  output = NULL,
  quietly = TRUE,
  size = "letter",
  cmd_options = NULL
)
```

Arguments

book	Book name. Currently only supports "The Historical Piecepacker".
gk	A game_kit R6 object.
output	Path to the output file. If NULL the function will guess a default.
quietly	Whether to hide document compilation output.
size	Paper size (either "letter", or "A4").
cmd_options	Options passed to pandoc when using non-pdf output formats. If NULL we try to guess a good set of options.

save_ruleset	<i>Save ruleset</i>
--------------	---------------------

Description

save_ruleset save ruleset of a game. save_pamphlet is a variant that saves the ruleset as a (tri-fold) pamphlet.

Usage

```

save_ruleset(
  game,
  gk = game_kit(),
  output = NULL,
  quietly = TRUE,
  size = "letter",
  ...,
  game_info = NULL,
  game_files = NULL,
  cmd_options = NULL
)

save_pamphlet(
  game,
  gk = game_kit(),
  output = NULL,
  quietly = TRUE,
  size = "letter",
  duplex_edge = "short",
  ...,
  game_info = NULL,
  game_files = NULL,
  save_promo_fn = save_promo_image
)

```

Arguments

game	Game name to generate ruleset for. See names_rulesets() . Will be normalized by ppdf::normalize_name() . If output is a pdf file may be a character vector of game names.
gk	A <code>game_kit</code> R6 object.
output	Path to the output file. If NULL the function will guess a default.
quietly	Whether to hide document compilation output.
size	Paper size (either "letter", or "A4").
...	Ignored
game_info	List with game info. If NULL then we use <code>yaml::yaml.load_file(system.file("extdata/game_info", package = "pprules"))</code> .
game_files	Character vector of (full path to) "Rtex" game rules. If NULL then we use <code>list.files(system.file("rules", package = "pprules"), full.names = TRUE)</code> .
cmd_options	Options passed to pandoc when using non-pdf output formats. If NULL we try to guess a good set of options.
duplex_edge	String specifying the desired duplex printing edge. If "short" match the second page along its short edge (second page flipped up, easier to preview on computer) and if "long" match along its long edge (second page flipped upside down, usual printer default).

`save_promo_fn` A function with arguments `game`, `gk`, and `file` that saves a promo image for `game`. Defaults to [save_promo_image\(\)](#).

Value

The path of the generated file invisibly.

Examples

```
cfg <- piecepackr::game_systems()$dual_piecepacks_expansion
gk <- game_kit(list(cfg = cfg))
if (pprules::has_xelatex()) {
  output <- tempfile(fileext = ".pdf")
  save_pamphlet("tablut", gk = gk, output = output)
  # xopen::xopen(output)
  # browseURL(output)
}
```

Index

`game_kit`, [2](#)

`names_rulesets`, [2](#)

`names_rulesets()`, [5](#)

`piecepackr::game_systems()`, [2](#)

`piecepackr::pp_cfg()`, [2](#)

`piecepackr::render_piece()`, [3](#)

`ppdf::normalize_name()`, [3](#), [5](#)

`ppdf::piecepack_games()`, [3](#)

`save_pamphlet (save_ruleset)`, [4](#)

`save_pamphlet()`, [3](#)

`save_promo_image`, [3](#)

`save_promo_image()`, [6](#)

`save_rulebook`, [4](#)

`save_ruleset`, [4](#)

`save_ruleset()`, [3](#)